Package 'GPSeqClus'

October 12, 2022

Type Package

Title Sequential Clustering Algorithm for Location Data

Version 1.3.0

Date 2022-07-29

Author Clapp, J. G.

Maintainer Justin Clapp <justin.clapp@wyo.gov>

Description Applies sequential clustering algorithm to animal location data based on user-defined parameters. Plots interactive cluster maps and provides a summary dataframe with attributes for each cluster commonly used as covariates in subsequent modeling efforts. Additional functions provide individual keyhole markup language plots for quick assessment, and export of global positioning system exchange format files for navigation purposes.
Methods can be found at <doi:10.1111/2041-210X.13572>.

Depends R (>= 3.5)

Imports geosphere, htmlwidgets, leaflet, leaflet.extras, plotKML, plyr, purrr, rgdal, sp, spacetime, stats, suncalc, tcltk, utils

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2022-08-09 14:20:08 UTC

R topics documented:

`addTitle` *label plots*

Description

label plots

Usage

```
addTitle(
  object,
  text,
  color = "black",
  fontSize = "20px",
  fontFamily = "Sans",
  leftPosition = 50,
  topPosition = 2
)
```

Arguments

`object`	leaflet plot
`text`	text
`color`	text color
`fontSize`	font size
`fontFamily`	font
`leftPosition`	left position
`topPosition`	top position

Value

title on plots

exp_clus_gpx *Export cluster .gpx file*

Description

Uses results from 'GPSeq_clus" to export .gpx file from specified AID and vector of desired cluster numbers for navigation during field site investigations.

Usage

```
exp_clus_gpx(AID, cn = "all", locs, cs, centroid_calc = "mean", dir = NULL)
```

Arguments

AID	Desired AID from sequential cluster output
cn	Numeric vector of desired cluster numbers to include in .gpx output, default is "all"
locs	Location dataframe output from GPSeq_clus()
cs	Cluster summary output from GPSeq_clus()
centroid_calc	'mean' (default) or 'median' centroid plot
dir	File path to save output

Value

.gpx file

Examples

```
exp_clus_gpx(AID = "ML1605M", cn = 4,
          locs = GPSeq_clus(dat = ML_ex_dat[1:50,], search_radius_m = 200, window_days = 6,
                    clus_min_locs = 3, show_plots = c(FALSE, "mean"))[[1]],
           cs = GPSeq_clus(dat = ML_ex_dat[1:50,], search_radius_m = 200, window_days = 6,
                  clus_min_locs = 3, show_plots = c(FALSE, "mean"))[[2]],
              dir= tempdir()
)
```

GPSeq_clus *Sequential cluster algorithm of location data*

Description

Applies sequential clustering algorithm to location data based on user-defined parameters and appends results to the dataframe. Provides a summary dataframe with attributes for each cluster commonly used as covariates in subsequent modeling efforts. Plots interactive cluster maps.

Usage

```
GPSeq_clus(
  dat,
  search_radius_m,
  window_days,
  clus_min_locs = 2,
  centroid_calc = "mean",
  show_plots = c(TRUE, "mean"),
  scale_plot_clus = TRUE,
  store_plots = FALSE,
  season_breaks_jul = NA,
  daylight_hrs = NA
)
```

Arguments

dat	Any dataframe including single or multiple animal location datasets that includes: **$AID** Animal identification for each location **$TelemDate** Location timestamps as POSIXct format "YYYY-MM-DD HH:MM:SS" with single "tzone" attribute **$Long** Longitude values as decimal degrees (-180 to +180) including NAs for failed fixes **$Lat** Latitude values as decimal degrees (-90 to +90) including NAs for failed fixes
search_radius_m	Search radius (meters) from cluster centroid when building clusters.
window_days	Temporal window (days) to search for new locations from the most recent location in a cluster
clus_min_locs	Minimum number of locations required to form a cluster. Default is 2.
centroid_calc	Method for recalculating centroids when actively building clusters - e.g., "median" or "mean" (default). Not to be confused with plotting the "mean" or "median" centroid once a cluster has been built.
show_plots	Vector of TRUE/FALSE for plotting followed by plotting argument for the "median" or "mean" centroid - e.g., c(TRUE, "mean") (default)

`scale_plot_clus`
: When plotting, scale cluster markers based on number of locations (TRUE/FALSE).

`store_plots`
: When plotting, also assign map outputs to global environment (TRUE/FALSE).

`season_breaks_jul`
: Ascending numeric vector of julian days (0-365) used to classify by season/parturition/hunting seasons etc. e.g., c(121, 274, 305) result may be: 1 Nov - 30 Apr (winter = 0), 1 May - 31 Aug (summer = 1), 1 Oct - 31 Oct (hunting season = 2)

`daylight_hrs`
: Manually set start and stop hours (0-24) to classify day and night locations. - e.g. c(6,18) would classify 6AM - 6PM as daylight hrs. NA (default) uses 'suncalc' package to convert cluster location and time to be classified based on specific specific sunrise and sunset times.

Value

Returns a list containing two dataframes. The first contains the original location dataframe with "clus_ID" column assigning each row a cluster ID if applicable. The second dataframe in the list contains a summary of sequential clusters and common cluster attributes (descriptions below) for subsequent modeling. If 'show_plots' argument is active, returns interactive maps of locations and clusters by animal.

AID Animal identification

clus_ID Sequential cluster ID number

clus_start Timestamp of first location in cluster

clus_end Timestamp of last location in cluster

clus_status "Closed" if the time window (window_days) has expired for the cluster according to users Sys.time() output. These clusters are therefore solidified and should not change if appending new location data. "Open" if the time window remains open at the time the function was run. "Open" clusters have the ability to shift sequence, combine with other clusters, emerge as a new cluster, etc. This attribute becomes relevant when appending new satellite data to the location dataframe, and may serve as an index of whether an animal continues to actively visit the cluster site within the time window.

g_c_Long Geometic centroid longitude value calculated using the mean

g_c_Lat Geometic centroid latitude value calculated using the mean

g_med_Long Geometic centroid longitude value calculated using the median

g_med_Lat Geometic centroid latitude value calculated using the median

clus_dur_hr Hours from the first to last locations of the cluster

n_clus_locs Number of locations within the cluster

visits Number of visits/revisits to the cluster based on the number of times locations fall outside the search radius and return to add locations to the cluster

fix_succ_clus_dur Fix rate success during the duration of the cluster

adj_clus_locs Adjusted number of cluster locations accounting for missed fixes (number cluster locations / fix success of cluster duration)

fid Fidelity to the cluster during cluster duration (number locations on cluster - number locations off cluster)

max_foray Maximum location distance (meters) from centroid during cluster duration for all locations

clus_radius Maximum location distance (meters) from centroid during cluster duration for cluster-attributed locations

avg_clus_dist Mean distance from all cluster locations to centroid

n_24_per Number of unique 24 hr periods during the cluster duration that hold at least one cluster location

bin_24hr Binary output for cluster duration (0 == less or equal to 24hr, 1 == greater than 24hr)

season Nominal attribute for user defined seasons based on 'season_breaks_jul' argument

night_pts Number of night cluster locations based on 'daylight_hrs' argument

night_prop Proportion of night cluster locations

Examples

```
GPSeq_clus(dat = ML_ex_dat[1:50,], search_radius_m = 200, window_days = 6,
           clus_min_locs = 3, show_plots = c(FALSE, "mean"))


GPSeq_clus(dat = ML_ex_dat, search_radius_m = 50, window_days = 2.5, clus_min_locs = 12,
        centroid_calc = "median", show_plots = c(TRUE, "median"), scale_plot_clus = FALSE,
           season_breaks_jul = c(120, 240, 300), daylight_hrs = c(8, 16))
```

---

ind_clus_kml *Plot individual cluster .kml*

Description

Uses results from 'GPSeq_clus" to plot individual cluster .kmls

Usage

```
ind_clus_kml(
  AID,
  cn,
  locs,
  cs,
  centroid_calc = "mean",
  overwrite = TRUE,
  dir = NULL
)
```

Arguments

AID	Desired AID from sequential cluster output
cn	Desired cluster number
locs	Location dataframe output from GPSeq_clus()
cs	Cluster summary output from GPSeq_clus()
centroid_calc	'mean' (default) or 'median' centroid plot
overwrite	TRUE (default) labels output as "ind.kml" that overwrites with each run within tempdir(). FALSE saves outputs as "AID_cn"
dir	File path when saving output

Value

Opens the cluster locations and centroid .kml for assessment.

Examples

```
ind_clus_kml(AID = "ML1605M", cn = 4,
         locs = GPSeq_clus(dat = ML_ex_dat[1:50,], search_radius_m = 200, window_days = 6,
                    clus_min_locs = 3, show_plots = c(FALSE, "mean"))[[1]],
          cs = GPSeq_clus(dat = ML_ex_dat[1:50,], search_radius_m = 200, window_days = 6,
                 clus_min_locs = 3, show_plots = c(FALSE, "mean"))[[2]]
)
```

---

julian_conv *Julian Conversion*

Description

Julian Conversion

Usage

```
julian_conv(x)
```

Arguments

x	vector of input dates

Value

vector of julian days

ML_ex_dat *Sample Data for Sequential Clustering Routine*

Description

A dataframe containing a subset of GPS location data from 2 male and 1 female mountain lions used for testing and running sequential cluster function examples.Example data provided by Wyoming Game and Fish Department, 2020.

Usage

```
ML_ex_dat
```

Format

A dataframe containing 4 columns:

AID animal identification

TelemDate location timestamp in POSIXct format

Lat latitude coordinates

Long longitude coordinates

moveMe *arrange columns*

Description

arrange columns

Usage

```
moveMe(data, tomove, where = "last", ba = NULL)
```

Arguments

data	input dataframe
tomove	which column(s) to move
where	where to move them - e.g. "before", "after", "first", "last"
ba	??

Value

Dataframe with new column order

Index